Gitについて

gitとは

分散型のファイル管理システムのこと。履歴を記録しながら、バージョン管理をするので、「あ、あの時のファイルって中身なんだったっけ」とかそういう時に便利。

分散型ってなんやねんとかそういうのは、このへん参照してください。

- https://github.com/masaru-b-cl/introduction-to-vcs
- https://github.com/takanabe/introduction-to-git/blob/master/01_what_is_git.md

Title

コミットメッセージの編集のvimコマンド

command	
`:w`	save
`:q`	quit

add

ファイルやディレクトリをIndexに登録するコマンド

`git add <filepattern>`

filepattern にはワイルドカードやオプションをつけて登録できます。

	内容
`*.txt`	.txtのファイルをすべて登録
`.`	サブディレクトリを含めたすべてのファイルを登録

以前にコミットした事のあるファイルを全てステージに登録したい
`git add -u`

branch

- `git merge branch_name` で、今選択されているブランチに branch_name という名前のブランチのコミット内容を取り込むことができる
- マージするときには「マージコミット」という特殊なコミットが行われ、マージコミットのコミットオブジェクトは親をふたつ持つ
- そのマージコミットが持つ「この瞬間のファイルの状態」は、ふたつの親が持っていた内容を合わせたものになる

- `git branch -d branch_name` とすることで、`branch_name` という名前のブランチを消すことができる

何かの変更を行うときには、新しいbranchを作成してから、そのbranchの中で変更を行うのが良い。

新しいブランチを作成してそちらに移動する

```
git checkout -b <new branch name>
```

特定の作業をするためのブランチを`<topic branch>`とか`<feature branch>`という

Merge

二つのブランチを統合するコマンドのこと。

Aというブランチの内容をBのブランチに反映させるときは

1. ブランチBを選択 `git checkout branch-b` して
2. ブランチAをmerge `git merge branch-a`

Fast-Foward と recursive strategy な merge

Fast-Forward: mergeする先と、現在のブランチの間で枝分かれが行われていない。
もし明示的にr-sなmergeをしたければ、オプション指定 `--no-ff` をつければ良い

```
git merge --no-ff <mergeするbranchの名前>
```

リモートリポジトリについて

- 手元のリポジトリでの変更内容を他のリポジトリに反映したり、その逆をするためには、リモートリポジトリとしてそのリポジトリを登録する必要がある。
  1. `git clone` でリポジトリ複製した場合
     - Gitが勝手に `Origin` という名前で登録する
  2. そうでない場合 `git remote add <name><path/to/repo.git>` で登録する
- リモートリポジトリに存在しているブランチを直接編集できない
- なので、リモートリポジトリにあるブランチを追跡するようなブランチを、自分のリポジトリ（ローカルリポジトリ）の中に作る必要がある。

ローカルで新しいブランチ `hoge_branch` を切ってリモートリポジトリにpushする

1. master（もしくは編集したいブランチ）から新しいブランチを作る

- `git checkout -b hoge_branch master`

2. 何かしら編集してコミットする

- `git add *.txt` and `git commit -m "Add text files"`

3. リモートリポジトリに、新しく作ったブランチを複製する

- `git push origin <local_branch_name>:<new_branch_name@remote>`
  - `git push origin hoge_branch:hoge_branch`
- 同じ名前でリモートに登録する場合は省略できる。
  - `git push origin hoge_branch`と同じ意味

1. 複製したブランチを、ローカルのブランチが追跡するように設定する

- `git branch --set-upstream-to=origin/hoge_branch hoge_branch`
- Note: 後半のリモートに登録して追跡することを一気にやるコマンド
  - `git push -u origin hoge_branch`

後始末

要らなくなったブランチは削除して、リモートリポジトリからも消さないといけない。順番としては

1. ローカルのブランチを削除する。
2. リモートリポジトリの消したいブランチを消す。

という手順。コードは以下のとおり。

```
git branch -d delete_branch
git push origin :delete_branch
```

二行目は、リモートリポジトリにある`delete_branch`を何もないブランチで上書きする。みたいな気持ち。

リモートにブランチを作る時の構文`git push <remote repo> <local branch>:<make branch name at remote>`を思い出せばなんとなくわからんでもない構文。

いま選択しているブランチをあるコミットの状態まで戻す

```
git reset --hard <戻りたいコミットのid>
```

追跡ブランチとは

追跡ブランチはリモート追跡ブランチ(remote-tracking branch)と上流ブランチ(upstream branch)という2つの概念からなる

リモート追跡ブランチ

- 場所：ローカルリポジトリ
- 役割：他のリポジトリの状態を追いかける
- `git fetch`で更新されるブランチのこと
- 普通の設定でリモートリポジトリを登録していれば`origin/master`とか`origin/foo`

上流ブランチ

- 引数なしで`git pull`したときの対象となるブランチ

- `git checkout master` and `git pull` した時、`master` は自動的に `origin/master` の変更を引っ張ってくる（mergeする）。
-